

# Circuits with Arbitrary Gates for Random Operators \*

S. Jukna      G. Schnitger

## Abstract

We consider boolean circuits computing  $n$ -operators  $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ . As gates we allow arbitrary boolean functions; neither fanin nor fanout of gates is restricted. An operator is linear if it computes  $n$  linear forms, that is, computes a matrix-vector product  $A\vec{x}$  over  $GF(2)$ .

We prove the existence of  $n$ -operators requiring about  $n^2$  wires in any circuit, and linear  $n$ -operators requiring about  $n^2/\log n$  wires in depth-2 circuits, if either all output gates or all gates on the middle layer are linear.

## 1 Introduction

We consider general circuits computing  $n$ -operators  $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ . As gates we allow *arbitrary* boolean functions of their inputs; there is no restriction on their fanin or fanout. Thus, the phenomenon which causes complexity of such circuits is *information transfer* rather than *information processing* as in the case of single functions. Such a circuit is a directed acyclic graph with  $n$  input nodes  $x_1, \dots, x_n$  and  $n$  output nodes  $y_1, \dots, y_n$ . Each non-input node computes some boolean function of its predecessors. A circuit computes an operator  $f = (f_1, \dots, f_n)$  if, for all  $i = 1, \dots, n$ , the boolean function computed at the  $i$ th output node  $y_i$  is the  $i$ th component  $f_i$  of the operator  $f$ . The *depth* of a circuit is the largest number of wires in a path from an input to an output node.

The *size* of a circuit is the total number of wires in it. We will denote by  $s_d(f)$  the smallest number of wires in a general circuit of depth at most  $d$  computing  $f$ . If there are no restrictions on the depth, the corresponding measure is denoted by  $s(f)$ . Note that  $s(f) \leq s_1(f) \leq n^2$  holds for any  $n$ -operator, so quadratic lower bounds are the highest ones.

Circuits of depth 2 constitute the first non-trivial model. Interest in depth-2 circuits comes from the following important result of Valiant [17]: If in every depth-2 circuit, computing  $f$  with  $O(n/\ln \ln n)$  gates on the middle layer, at least  $n^{1+\Omega(1)}$  direct wires must connect inputs with output gates, then  $f$  cannot

---

\*Research of both authors supported by a DFG grant SCHN 503/4-1. University of Frankfurt, Institute of Computer Science, Frankfurt am Main, Germany

be computed by log-depth circuits with a linear number of fanin-2 gates. To prove a super-linear lower bound for log-depth circuits is an old and well-known problem in circuit complexity.

Super-linear lower bounds up to  $s_2(f) = \Omega(n \log^2 n)$  were proved using graph-theoretic arguments by analyzing some super-concentration properties of the circuit as a graph [5, 9, 10, 12, 11, 1, 13, 14, 15]. Higher lower bounds of the form  $s_2(f) = \Omega(n^{3/2})$  were recently proved using information theoretical arguments [4, 6]. For larger depth  $d$  known lower bounds are only slightly non-linear. All these bounds, however, are on the *total* number of wires, so they still have no consequences for log-depth circuits.

In fact, in the class of general circuits, even the question about the complexity of a *random* operator remained unclear. In particular, it was unclear whether operators requiring a quadratic number of wires (even in depth 2) exist at all?

## 2 Circuits for general operators

Note that a direct counting argument, as in the case of constant fanin circuits, does not work for general circuits: already for  $d > n + \log n$ , the number  $2^{2^d}$  of possible boolean functions that may be assigned to a node of fanin  $d$  may be larger than the total number  $2^{n^{2^n}}$  of  $n$ -operators.

Our first result is an observation that this bad situation can be excluded by just turning the power of circuits against themselves to ensure that, in an optimal circuit, no gate can have fanin larger than  $n$ . This leads us to

**Theorem 1.** *For almost all  $n$ -operators  $f$ ,  $s(f) = \Omega(n^2)$ .*

*Proof.* Let  $\mu(L)$  be the number of different  $n$ -operators computable by boolean circuits with at most  $L$  wires. Our goal is to upper bound this number in terms of  $n$  and  $L$ , and compare this bound with the total number  $2^{n^{2^n}}$  of  $n$ -operators.

Take an optimal circuit with  $\ell \leq L$  wires computing some  $n$ -operator; hence,  $\ell \leq n^2$ . Then  $\ell = \sum_{i=1}^m d_i$ , where  $d_1, \dots, d_m$  are the fanins of its gates. It is clear that we need  $m \geq n$  gates, since we must have  $n$  input gates. On the other hand,  $m \leq \ell + n + 2 \leq 2n^2$  gates are always enough since every non-input gate, besides two possible constant gates, must have nonzero fanin.

We now make use of the fact that the gates in our circuits may be *arbitrary* boolean functions: This allows us to assume that  $d_i \leq n$  for all  $i$ . Indeed, if  $d_i > n$ , then we can replace the  $i$ th gate by the boolean function computed at this gate and join it to all  $n$  input variables; when doing this, the total number of wires in the circuit can only decrease.

The number of sequences  $d_1, \dots, d_m$  of fanins with  $0 \leq d_i \leq n$  does not exceed  $(n+1)^m$ . For each such sequence and for each  $i = 1, \dots, m$ , there are at most  $\binom{m}{d_i} \leq m^{d_i}$  possibilities to choose the set of inputs for the  $i$ th node and at

most  $2^{2^{d_i}}$  possibilities to assign a boolean function to this node. Hence,

$$\mu(L) \leq (n+1)^m \prod_{i=1}^m m^{d_i} \prod_{i=1}^m 2^{2^{d_i}} = (n+1)^m m^{\sum_{i=1}^m d_i} 2^{\sum_{i=1}^m 2^{d_i}}.$$

Since  $\sum_{i=1}^m d_i \leq L \leq n^2$  and  $m \leq 2n^2$ , this yields

$$\log_2 \mu(L) \leq \sum_{i=1}^m 2^{d_i} + O(n^2 \log_2 n).$$

We now observe that at most  $n/2$  nodes can have fanin larger than  $2L/n$ , for otherwise we would have more than  $(2L/n) \cdot (n/2) = L$  wires in total. Since  $m \leq 2n^2$  and since the fanin of each gate does not exceed  $n$ , we obtain that

$$\sum_{i=1}^m 2^{d_i} \leq (m - n/2)2^{2L/n} + (n/2)2^n \leq 2n^2 4^{L/n} + n2^{n-1}.$$

Hence,

$$\log_2 \mu(L) \leq 2n^2 4^{L/n} + n2^{n-1} + O(n^2 \log_2 n). \quad (1)$$

Since the total number of operators  $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$  is  $2^{n2^n}$ , the smallest number  $L$  of wires sufficient to compute all of them must satisfy  $\log_2 \mu(L) \geq n2^n$ . By (1), this implies

$$2n^2 4^{L/n} \geq n2^{n-1} - O(n^2 \log_2 n).$$

Dividing both sides by  $2n^2$ , we obtain that  $4^{L/n} = \Omega(2^n/n)$ , and hence,  $L = \Omega(n^2)$ .  $\square$

### 3 Circuits for linear operators

An important class of operators are *linear* ones. Each such operator computes  $n$  linear forms, that is, computes a matrix-vector product  $f_A(\vec{x}) = A\vec{x}$  over  $GF(2)$  where  $A$  is an  $n \times n$   $(0, 1)$ -matrix. We are interested in the complexity  $s_2(f_A)$  of such operators in the class of depth-2 circuits.

If all gates are required to be *linear* (parities and their negations), then easy counting shows that some linear operators require  $\Omega(n^2/\log n)$  wires. It is also known that  $O(n^2/\log n)$  are also sufficient to compute any linear operator [16, 3, 2].

But what if we allow arbitrary (non-linear) boolean functions as gates—can we then compute linear operators  $f_A$  more efficiently? The largest known lower bound for an *explicit* linear operator  $f_A$  has the form  $s_2(f_A) = \Omega(n \log n)$  [11]. This raises the following question: Do *linear*  $n$ -operators requiring  $s_2(f_A) = \Omega(n^2/\log n)$  wires exist at all? We are only able to answer this question *positively* under the additional restriction that either all output gates of all gates on the middle layer must be linear functions.

The next theorem shows that the non-linearity of *middle* gates is no problem: any such circuit can be transformed into a *linear* circuit with almost the same number of wires. Hence, some linear  $n$ -operators require about  $n^2/\log n$  wires in such circuits.

**Theorem 2.** *If a depth-2 circuit computes a linear  $n$ -operator and only has linear gates on the output layer, then it can be transformed to an equivalent linear circuit by adding at most  $2n$  new wires.*

*Proof.* Let  $A$  be an  $n$ -by- $n$   $(0,1)$ -matrix, and let  $\Phi$  be a depth-2 circuit computing  $A\vec{x}$ . We may assume, for simplicity, that there are no direct wires from inputs to outputs: this can be easily achieved by adding  $n$  new wires on the first level. Assume that all output gates of  $\Phi$  are linear boolean functions. By adding one constant-1 function on the middle layer and at most  $n$  new wires on the second level, we can also assume that each output gate computes just the sum modulo 2 of its inputs (and not the negation of this sum).

Let  $h = (h_1, \dots, h_r) : \{0,1\}^n \rightarrow \{0,1\}^r$  be the operator computed by the gates on the middle layer. Since  $A\vec{0} = \vec{0}$  and each output gate computes the sum modulo 2 of its inputs, we may assume that  $h(\vec{0}) = \vec{0}$  as well: If  $h_j(\vec{0}) = 1$  for some  $j$ , then replace the function  $h_j$  by the function  $h'_j$  such that  $h'_j(\vec{0}) = 0$  and  $h'_j(\vec{x}) = h_j(\vec{x})$  for all  $\vec{x} \neq \vec{0}$ .

Let  $B$  be the  $n$ -by- $r$  adjacency  $(0,1)$ -matrix of the bipartite graph formed by the wires joining the gates on the middle layer with those on the output layer. Then  $A\vec{x} = B \cdot h(\vec{x})$  for all  $\vec{x} \in \{0,1\}^n$ . Write each vector  $\vec{x} = (x_1, \dots, x_n)$  as the linear combination  $\vec{x} = \sum_{i=1}^n x_i \vec{e}_i$  of unit vectors  $\vec{e}_1, \dots, \vec{e}_n \in \{0,1\}^n$ , and replace the operator  $h$  computed on the middle layer by a *linear* operator

$$h'(\vec{x}) := \sum_{i=1}^n x_i h(\vec{e}_i) \bmod 2.$$

Hence,  $h'(\vec{x}) = \vec{x}^\top M$ , where  $M$  is an  $n \times r$  matrix with rows  $h(\vec{e}_1), \dots, h(\vec{e}_n)$ . Using the linearity of the matrix-vector product, we obtain that (with all sums mod 2):

$$B \cdot h(\vec{x}) = A \cdot \left( \sum x_i \vec{e}_i \right) = \sum x_i A \vec{e}_i = \sum x_i B \cdot h(\vec{e}_i) = B \cdot h'(\vec{x}).$$

Hence, the new (linear) circuit  $\Phi'$  computes  $A\vec{x}$  as well. It remains to show that the number of wires in  $\Phi'$  does not exceed the number of wires in  $\Phi$ .

The wires on the second level haven't changed at all. To show that the *number* of wires on the first level has not increased as well, let  $\text{fanout}(x_i)$  be the fanout of the  $i$ th input node  $x_i$ , and  $\text{fanin}(h_j)$  the fanin of the  $j$ th gate  $h_j$  on the middle layer. Then  $\sum_{i=1}^n \text{fanout}(x_i) = \sum_{j=1}^r \text{fanin}(h_j)$  is the total number  $L$  of wires on the first level. We know that  $h(\vec{0}) = \vec{0}$ , that is,  $h_j(\vec{0}) = 0$  for all  $j = 1, \dots, r$ . Now we make a simple (but crucial) observation: if there is no wire from  $x_i$  to  $h_j$ , then  $h_j(\vec{e}_i) = h_j(\vec{0}) = 0$ . This implies that the  $j$ th column of  $M$  can have at most  $\text{fanin}(h_j)$  ones. Since the number of wires on the first level of  $\Phi'$  is just the total number of 1's in  $M$ , we are done.  $\square$

The second case—when only gates on the middle layer are required to be linear—is more delicate. That such circuits *can* be more powerful than linear ones, was shown in [7]. Given a boolean  $n \times n$  matrix  $A$ , say that a circuit *weakly computes* the operator  $f_A(\vec{x}) = A\vec{x}$  if it correctly computes it on all  $n$  unit vectors  $\vec{e}_1, \dots, \vec{e}_n$ . Note that, for *linear* circuits, this is no relaxation: such a circuit weakly computes  $f_A$  iff it correctly computes  $f_A$  on all inputs. Hence, some linear operators cannot be weakly computed by *linear* depth-2 circuits using fewer than  $\Omega(n^2/\log n)$  wires. It is however shown in [7] that the situation changes drastically if we only use linear gates on the middle layer but allow non-linear gates on the output layer, then *any* linear  $n$ -operator can be weakly computed using only  $O(n \log n)$  wires.

Still, using Kolmogorov complexity arguments, we can prove that, for some matrices  $A$ , such circuits require a quadratic number of wires to compute the entire operator  $A\vec{x}$ .

**Theorem 3.** *If middle gates are required to be linear, then linear  $n$ -operators  $f_A$  with  $s_2(f_A) = \Omega(n^2/\log n)$  exist.*

*Proof.* We use the Kolmogorov complexity argument known as the *incompressibility argument* (see [8] for background). Since we have  $2^{n^2}$  matrices, some matrix  $A$  requires  $n^2$  bits to describe it. Hence, the linear operator  $f_A(\vec{x}) = A\vec{x}$  cannot be described using fewer than  $n^2 - O(1)$  bits, as well.

Fix an arbitrary depth-2 circuit  $\Phi$  computing  $f_A$ , and assume that all its gates on the middle layer are linear. Let  $L$  be the number of wires in  $\Phi$ . As before, we may assume that there are no direct wires from inputs to outputs. Our goal is to show that, using the circuit  $\Phi$ , the operator  $f_A$  can be described using  $O(L \log n)$  bits. This will imply the desired lower bound  $L = \Omega(n^2/\log n)$  on the number of wires.

Let  $r$  be the number of nodes on the middle layer of  $\Phi$ . Since at these nodes only linear functions are computed, the first level (between inputs and middle layer) computes some linear operator  $\vec{y} = B\vec{x}$ , where  $B$  is the  $r$ -by- $n$  adjacency matrix of the bipartite graph formed by the wires joining the gates on the input layer with those on the middle layer. Let also  $C$  be the  $n$ -by- $r$  adjacency matrix of the bipartite graph formed by the wires joining the gates on the middle layer with those on the output layer. Hence,  $L = |B| + |C|$  where  $|B|$  denotes the number of 1s in  $B$ .

Using these two matrices  $B$  and  $C$  as well as the fact that the operator computed by the circuit  $\Phi$  is linear, we can encode this operator using  $O(L \log n)$  bits as follows.

- Since  $|B| + |C| = L$ , both matrices  $B$  and  $C$  can be described using  $O(L \log n)$  bits, just by describing the positions of their 1-entries.
- The  $i$ th output gate of  $\Phi$  computes  $g_i(B\vec{x})$ , where  $g_i : \{0, 1\}^r \rightarrow \{0, 1\}$  is some boolean function depending only on rows of  $B$  seen by this gate, that is, on rows corresponding to the  $d_i$  nodes on the middle layer seen by this gate. Let  $B_i$  be the  $d_i \times n$  submatrix of  $B$  formed by these rows.

Let  $\text{Im}(B_i) = \{B_i \vec{x} : \vec{x} \in \{0, 1\}^n\}$  be the column space of  $B_i$ . If this space has dimension  $t$  then any  $t$  linearly independent columns of  $B$  form its basis. Take the set  $B'_i = \{\vec{u}_1, \dots, \vec{u}_t\}$  of the first  $t$  linearly independent columns of  $B_i$ , and call it the *first basis* of  $\text{Im}(B_i)$ .

- Encode the behavior of  $g_i$  on this basis  $B'_i$  by the string  $g_i(\vec{u}_1), \dots, g_i(\vec{u}_t)$  of  $t \leq d_i$  bits. The entire string, for all  $n$  output gates  $g_1, \dots, g_n$ , has length at most  $\sum_{i=1}^n d_i \leq L$ .

Having this encoding, we can recover the value  $g_i(\vec{x})$  of the  $i$ th output gate on a given input  $\vec{x} \in \{0, 1\}^n$  as follows.

1. Compute  $\vec{y}_i = B_i \vec{x}$ . We can do this since the  $i$ th row of  $C$  tells us what rows of  $B$  appear in  $B_i$ , and we know the entire matrix  $B$ .
2. Take the first basis  $B'_i$  of  $\text{Im}(B_i)$  and write  $\vec{y}_i$  as a linear combination  $\vec{y}_i = \sum_{k=1}^t \lambda_k \vec{u}_k$  of basis vectors over  $GF(2)$ .
3. Give  $z_i = \sum_{k=1}^t \lambda_k g_i(\vec{u}_k) \bmod 2$  as an output. We can compute this number since we know the values  $g_i(\vec{u}_1), \dots, g_i(\vec{u}_t)$ .

Since the circuit computes  $A\vec{x}$ , the  $i$ th output gate must compute the scalar product  $\langle \vec{a}_i, \vec{x} \rangle$  of input vector  $\vec{x}$  with the  $i$ th row  $\vec{a}_i$  of  $A$ . Hence,  $g_i(B\vec{x}) = \langle \vec{a}_i, \vec{x} \rangle$ , meaning that  $g_i$  must be *linear* on  $\text{Im}(B)$ . Since  $g_i$  can only see the middle gates corresponding to the rows of  $B_i$ , this implies that  $g_i$  must be linear also on  $\text{Im}(B_i)$ . Thus,

$$z_i = \sum_{k=1}^t \lambda_k g_i(\vec{u}_k) = g_i\left(\sum_{k=1}^t \lambda_k \vec{u}_k\right) = g_i(\vec{y}_i) = g_i(B_i \vec{x}) = g_i(B\vec{x}),$$

that is,  $z_i$  is a scalar product of  $\vec{x}$  with the  $i$ th row of  $A$ , as desired. □

## 4 Conclusion

We have shown that, even when arbitrary boolean functions can be used as gates, some operators  $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$  require about  $n^2$  wires. We have also shown that some *linear* operators require about  $n^2 / \log n$  wires in depth-2 circuits, if either all output gates or all gates on the middle layer are required to be linear.

We conjecture that the same lower bound for depth-2 circuits computing linear operators should also hold without any restrictions on used gates.

## References

- [1] N. Alon, P. Pudlák, Superconcentrators of depth 2 and 3; odd levels help (rarely), J. Comput. Sys. Sci. 48 (1994) 194–202.

- [2] N. Alon, M. Karchmer, A. Wigderson, Linear circuits over  $\text{GF}(2)$ , *SIAM J. Comput.* 19(6) (1990) 1064–1067.
- [3] S. Bublitz, Decomposition of graphs and monotone size of homogeneous functions, *Acta Inform.* 23 (1986) 689–696.
- [4] D. Y. Cherukhin, The lower estimate of complexity in the class of schemes of depth 2 without restrictions on a basis, *Moscow Univ. Math. Bull.* 60(4) (2005) 42–44.
- [5] D. Dolev, C. Dwork, N. Pippenger, A. Wigderson, Superconcentrators, generalizer and generalized connectors with limited depth, in: *Proc. 15th STOC* (1983), pp. 42–51.
- [6] S. Jukna, Entropy of operators or why matrix multiplication is hard for depth-two circuits, *Theory of Comp. Syst.* 46(2) (2010) 301–310.
- [7] S. Jukna, Representing  $(0,1)$ -matrices by depth-2 circuits with arbitrary gates, *Discrete Math.* 310 (2010) 184–187.
- [8] M. Li, P. Vitáni, *An Introduction to Kolmogorov Complexity and Its Applications*, 2nd Edition, Springer-Verlag, 1997.
- [9] N. Pippenger, Superconcentrators, *SIAM J. Comput.* 6 (1977) 298–304.
- [10] N. Pippenger, Superconcentrators of depth 2, *J. Comput. Syst. Sci.* 24 (1982) 82–90.
- [11] P. Pudlák, Communication in bounded depth circuits, *Combinatorica* 14 (2) (1994) 203–216.
- [12] P. Pudlák, P. Savický, On shifting networks, *Theoret. Comput. Sci.* 116 (1993) 415–419.
- [13] P. Pudlák, V. Rödl, J. Sgall, Boolean circuits, tensor ranks, and communication complexity, *SIAM J. Comput.* 26(3) (1997) 605–633.
- [14] J. Radhakrishnan, A. Ta-Shma, Bounds for dispersers, extractors, and depth-two superconcentrators, *SIAM J. Discrete Math.* 13(1) (2000) 2–24.
- [15] R. Raz, A. Shpilka, Lower bounds for matrix product in bounded depth circuits with arbitrary gates, *SIAM J. Comput.* 32(2) (2003) 488–513.
- [16] Zs. Tuza, Coverings of graphs by complete bipartite subgraphs, complexity of 0-1 matrices, *Combinatorica* 4(1) (1984) 111–116.
- [17] L. Valiant, Graph-theoretic methods in low-level complexity, in *Proc. 6th MFCS*, Springer Lect. Notes in Comput. Sci. 53 (1977), pp. 162–176.